

HIGHLY SCALABLE SYSTEM AND METHOD OF REGULATING  
INTERNET TRAFFIC TO SERVER FARM TO SUPPORT (MIN,MAX)  
BANDWIDTH USAGE-BASED SERVICE LEVEL AGREEMENTS

BACKGROUND OF THE INVENTION

5 1. FIELD OF THE INVENTION

005040 " 040500 0343307 024550

The present invention generally relates to the global Internet and Internet World Wide Web (WWW) sites of various owners that are hosted by a service provider using a group of servers that are intended to meet established service levels. More particularly, this invention relates to a highly scalable system and method for supporting (min,max) based service level agreements on outbound bandwidth usage for a plurality of customers by regulating inbound traffic coming to a server farm where the server farm is comprised of numerous servers.

10

2. DESCRIPTION OF THE PRIOR ART

The Internet is the world's largest network and has become essential to businesses as well as to consumers. Many businesses have started outsourcing their e-business and e-commerce Web sites to service providers instead of running their Web sites on their own server(s) and managing them by themselves. Such a service provider needs to install a collection of servers (called a Web Server Farm (WSF) or a Universal Server Farm (USF)), which can be used by many different businesses to support their e-commerce and e-business. These business customers (the service provider's "customers") have different "capacity" requirements for their Web sites. Web Server Farms are connected to the Internet via high speed communications links such as T3 and OCx links. These links are shared by all of the Web sites and all of the users accessing the services hosted by the Web Server Farm. When businesses

15

20

(hereafter referred to as customers of a server farm, or customers) outsource their e-commerce and/or e-business to a service provider, they typically need some assurance as to the services they are getting from the service provider for their sites. Once the service provider has made a commitment to a customer to provide a certain level of service (called a Service Level Agreement (SLA)), the provider needs to maintain that level of service to that customer.

A general SLA on communications link bandwidth usage for a customer can be denoted by a pair of bandwidth constraints: the minimum guaranteed bandwidth,  $B_{min}(i,j)$ , and the maximum bandwidth bound,  $B_{max}(i,j)$ , for each  $i^{th}$  customer's  $j^{th}$  type or class traffic. The minimum (or min) bandwidth  $B_{min}(i,j)$  is a guaranteed bandwidth that the  $i^{th}$  customer's  $j^{th}$  type traffic will receive regardless of the bandwidth usage by other customers. The maximum (or max) bandwidth  $B_{max}(i,j)$  is an upper bound on the bandwidth that the  $i^{th}$  customer's  $j^{th}$  type traffic may receive provided that some unused bandwidth is available. Therefore, the range between  $B_{min}(i,j)$  and  $B_{max}(i,j)$  represents the bandwidth provided on an "available" or "best-effort" basis, and it is not necessarily guaranteed that the customer will obtain this bandwidth. In general, the unit cost to use the bandwidth up to  $B_{min}(i,j)$  is less than or equal to the unit cost to use the bandwidth between  $B_{min}(i,j)$  and  $B_{max}(i,j)$ . Such a unit cost assigned to one customer may differ from those assigned to other customers.

In the environment of Web site hosting, where communications link(s) between the Internet and a server farm is shared by a number of customers (i.e., traffic to and from customer Web sites share the communications link(s)), the bandwidth management on the outbound link, i.e., the link from a server farm to the Internet, is more important than the bandwidth management on the inbound link since the amount of traffic on the outbound link is many magnitudes greater than that on the inbound link. Furthermore, in most cases, the inbound traffic to the server farm is directly

responsible for the outbound traffic generated by the server farm. Therefore, the constraints  $B_{min}(i,j)$  and  $B_{max}(i,j)$  imposed by a service level agreement are typically applied to the outbound link bandwidth usage.

There are two types of bandwidth control systems that have been proposed either in the market or in the literature. One type is exemplified by the Access Point (AP) products from Lucent/Xedia ([www.xedia.com](http://www.xedia.com)) or by the Speed-Class products from PhaseCom ([www.speed-demon.com](http://www.speed-demon.com)). These products are self-contained units and they can be applied to regulate the outbound traffic by dropping some outbound packets to meet with the (minimum,maximum) bandwidth SLA for each customer. The other type of bandwidth control system is exemplified by U.S. Patent Application Serial No. [Attorney's Docket No. YO999-374], commonly assigned with the present invention. This system, referred to as Communications Bandwidth Management (CBM), operates to keep the generated outbound traffic within the SLAs by regulating the inbound traffic that is admitted to a server farm. As with the first type of bandwidth control system exemplified by AP and Speed-Class products, each CBM is a self-contained unit.

Bandwidth control systems of the types exemplified by Lucent AP noted above can be applied to enforce SLAs on the outbound link usage by each customer (and on each customer traffic type). Some of these systems are limited to supporting the minimum bandwidth SLA while others are able to support the (minimum,maximum) bandwidth SLA. A disadvantage with systems that enforce the outbound bandwidth SLA by dropping packets already generated by the server farm is that they induce undesirable performance instability. That is, when some outbound packets must be dropped, each system drops packets randomly, thus leading to frequent TCP (Transmission Control Protocol) retransmission, then to further congestion and packet dropping and eventually to thrashing and slowdown. The CBM system noted above solves such a performance instability problem by not admitting

inbound traffic whose output cannot be delivered due to exceeding the SLA. The major problem of AP and CBM units is that their scalability is limited. A large server farm requires more than one unit of AP or CBM unit. However, because each of these units is self-contained and standalone, they cannot collaborate to handle the amount of traffic beyond the capacity of a single unit. When a multiple number ("n") of AP or CBM systems are needed to be deployed to meet the capacity requirement, each unit will handle (1/n)-th of the total bandwidth or traffic, and therefore the sharing of the available bandwidth and borrowing of unused bandwidth among customers becomes impossible.

From the above, it can be seen that it would be desirable if a system for bandwidth control of a server farm were available that overcomes the scalability problem while eliminating the performance and bandwidth sharing shortcomings of the prior art.

#### SUMMARY OF THE INVENTION

The present invention provides a highly scalable system and method for guaranteeing and delivering (minimum,maximum) based communications link bandwidth SLAs to customers whose applications (e.g., Web sites) are hosted by a server farm that consists of a very large number of servers, e.g., hundreds of thousands of servers. The system of this invention prevents any single customer (or class of) traffic from "hogging" the entire bandwidth resource and penalizing others. The system accomplishes this in part through a feedback system that enforces the outbound link bandwidth SLAs by regulating the inbound traffic to a server farm. In this manner, the system of this invention provides a method by which differentiated services can be provided to various types of traffic, the generation of output from a server farm is avoided if that output cannot be delivered to end users, and any given objective function is optimized when allocating bandwidth beyond the minimums.

The system accomplishes its high scalability by allowing the deployment of more than one very simple inbound traffic limiter (or regulator) that performs the rate-based traffic admittance and by using a centralized rate scheduling algorithm. The system also provides means for any external system or operator to further limit the rates used by inbound traffic limiters.

According to industry practice, customers may have an SLA for each type or class of traffic, whereby (minimum,maximum) bandwidth bounds are imposed in which the minimum bandwidth represents the guaranteed bandwidth while the maximum bandwidth represents the upper bound to the as-available use bandwidth. The bandwidth control and management system of this invention enforces the outbound link bandwidth SLAs by regulating (thus limiting when needed) the various customers' inbound traffic to the server farm. Incoming traffic (e.g., Internet Protocol (IP) packets) can be classified into various classes/types (denoted by  $(i,j)$ ) by examining the packet destination address and the TCP port number. For each class  $(i,j)$ , there is a "target" rate denoted as  $R_t(i,j)$ , which is the amount of the  $i^{\text{th}}$  customer's  $j^{\text{th}}$  type traffic that can be admitted within a given service cycle time to the server farm which supports the  $i^{\text{th}}$  customer (this mechanism is known as the rate-based admittance). A centralized device is provided that computes  $R_t(i,j)$  using the history of admitted inbound traffic to the server farm, the history of rejected (or dropped) inbound traffic, the history of generated outbound traffic from the server farm, and the SLAs. Each dispatcher can use any suitable algorithm to balance the work load to servers when dispatching traffic to servers. Once  $R_t(i,j)$  values have been computed by the centralized device, the centralized device relays the  $R_t(i,j)$  values to the one or more elements (called inbound traffic limiters) that regulate the inbound traffic using the rates  $R_t(i,j)$  in a given service cycle time. The above process of computing and deploying  $R_t(i,j)$  values is repeated periodically. This period can be as often as the service cycle time.

In addition to enforcing the outbound link bandwidth SLAs in a highly

scalable fashion, other preferred feature of the present invention is the ability to control the computation of  $Rt(i,j)$  via an external means such as an operator and any server resource manager. Yet other preferred features of the present invention are the ability to distribute monitoring and traffic limiting functions even to each individual server level. Any existing workload dispatching product(s) can be used with this invention to create a large capacity dispatching network. Yet other preferred features of the present invention are the capabilities to regulate inbound traffic to alleviate the congestion (and thus performance) of other server farm resources (in addition to the outbound link bandwidth) such as web servers, data base and transaction servers, and the server farm intra-infrastructure. These are achieved by providing "bounds" to the centralized device.

Other objects and advantages of this invention will be better appreciated from the following detailed description.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 represents a system environment in which Internet server farm traffic is to be controlled and managed with a bandwidth control system in accordance with the present invention.

Figure 2 schematically represents a bandwidth control system operating within the system environment of Figure 1 in accordance with the present invention.

Figures 3 and 4 schematically represent two embodiments for an inbound traffic dispatching network represented in Figure 2.

Figure 5 schematically represents an inbound traffic limiter algorithm for use with inbound traffic limiters of Figures 2 through 4 and 7.

Figure 6 schematically represents a rate scheduling algorithm for computing  $R_t(i,j)$  with an inbound traffic scheduler unit represented in Figure 2.

Figure 7 represents an inbound traffic limiting system operating within each server in accordance with another embodiment of the present invention.

## 5 DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Figure 1 schematically represents a system environment in which traffic through an Internet server farm 10 can be regulated with a bandwidth control system in accordance with the present invention. The Internet server farm 10 is represented as being comprised of an inbound traffic (or TCP connection request) dispatching network 12 that dispatches inbound traffic 14 to appropriate servers 16. The invention is intended for use with a very large number of servers 16 (the size beyond the capacity of a single dispatcher unit) of potentially different capacities that create the outbound traffic 18 of the server farm 10. An objective of this invention is to provide a highly scalable system and method that manages the outbound bandwidth usage of various customers (and thus customer traffic) subject to (min,max) bandwidth-based service level agreements (SLAs) by regulating the inbound traffic 14 of various customers. Table 1 contains a summary of symbols and notations used throughout the following discussion.

TABLE 1

20	i	The $i^{\text{th}}$ customer.
	j	The $j^{\text{th}}$ traffic type/class.
	k	The $k^{\text{th}}$ server.
	$R_a(i,j,k)$	Inbound traffic of the $j^{\text{th}}$ type of the $i^{\text{th}}$ customer that has been admitted at the $k^{\text{th}}$ server.
25	$R_a(i,j)$	The total inbound traffic of the $j^{\text{th}}$ type of the $i^{\text{th}}$ customer that

		has been admitted; equivalent to the sum of $R_a(i,j,k)$ over all $k$ .
	$R_r(i,j,k)$	Inbound traffic of the $j^{\text{th}}$ type of the $i^{\text{th}}$ customer that has been rejected at the $k^{\text{th}}$ server.
5	$R_r(i,j)$	The total inbound traffic of the $j^{\text{th}}$ type of the $i^{\text{th}}$ customer that has been rejected; equivalent to the sum of $R_p(i,j,k)$ over all $k$ .
	$R_t(i,j,k)$	The allowable (target) traffic rate for the $i^{\text{th}}$ customer's $j^{\text{th}}$ type traffic at the $k^{\text{th}}$ server.
	$R_t(i,j)$	The total allowable traffic rate for the $i^{\text{th}}$ customer's $j^{\text{th}}$ type traffic, equivalent to the sum of $R_t(i,j,k)$ over all $k$ .
10	$B(i,j,k)$	The $i^{\text{th}}$ customer's $j^{\text{th}}$ type outbound traffic from the $k^{\text{th}}$ server.
	$B(i,j)$	The total of the $i^{\text{th}}$ customer's $j^{\text{th}}$ type outbound traffic, equivalent to the sum of $B(i,j,k)$ over all $k$ .
	$b(i,j)$	The expected bandwidth usage by a unit of inbound traffic type $(i,j)$ .
15	$C(i,j,k)$	The server resource (processing capacity) that is allocated to the $i^{\text{th}}$ customer's $j^{\text{th}}$ type traffic at the $k^{\text{th}}$ server.
	$C(i,j)$	The total processing capacity that is allocated to the $i^{\text{th}}$ customer's $j^{\text{th}}$ type traffic, equivalent to the sum of $C(i,j,k)$ over all $k$ .
20	$c(i,j)$	The expected server resource usage by a unit of $(i,j)$ traffic.
	$B_{\min}(i,j)$	The guaranteed outbound bandwidth usage on the $i^{\text{th}}$ customer's $j^{\text{th}}$ type traffic.
	$B_{\max}(i,j)$	The maximum on the outbound bandwidth usage on the $i^{\text{th}}$ customer's $j^{\text{th}}$ type traffic.
25	$B_{\text{total}}$	The total usable bandwidth available for allocation.
	$R_{\text{bound}}(i,j)$	An optional bound on $R_t(i,j)$ that may be set manually or by any other resource manager of a server farm.

Figure 2 schematically illustrates an embodiment of the invention



operating within the system environment shown in Figure 1. A unit referred to herein as the inbound traffic scheduler (ITS) unit 20 is employed to observe the amount of incoming traffic 14 that consists of the amount of admitted inbound traffic and the amount of rejected traffic. The inbound traffic dispatching network 12 monitors both admitted and rejected traffic amount. The ITS unit 20 also observes outbound traffic 18. The ITS unit 20 then computes the expected amount of outbound traffic that would be generated when one unit of traffic is admitted to a server 16, computes the inbound traffic target rates, and informs the rates to an inbound traffic limiter (ITL) 22. The ITL 22 then regulates the arriving inbound traffic 14 by imposing target rates at which inbound traffic 14 is admitted. Each of these functions is performed for the  $i^{\text{th}}$  customer's  $j^{\text{th}}$  class traffic within a service cycle time, which is a unit of time or period that is repeated. Optionally observed by the ITS unit 20 is the average resource usage  $c(i,j)$  by a unit of type  $(i,j)$  inbound traffic 14.

As indicated in Table 1,  $R_a(i,j)$  denotes the amount of inbound traffic 14 admitted and  $R_r(i,j)$  denotes the amount of inbound traffic 14 rejected. Both are obtained by the ITS unit 20 during a service cycle time (thus representing a rate) for the  $i^{\text{th}}$  customer's  $j^{\text{th}}$  class traffic.  $R_r(i,j)$  greater than zero implies that the  $R_r(i,j)$  amount of traffic was rejected due to inbound traffic 14 exceeding the usage of the agreed upon outbound bandwidth.  $R_t(i,j)$  denotes the allowable (thus targeted) portion of inbound traffic 14 within a service cycle time for the  $i^{\text{th}}$  customer's  $j^{\text{th}}$  class traffic. Here,  $R_a(i,j)$  is smaller than or equal to  $R_t(i,j)$  as a result of the operation of the ITL 22.  $B(i,j)$  denotes the total amount of outbound traffic 18 generated for the  $i^{\text{th}}$  customer's  $j^{\text{th}}$  class traffic within a service cycle time, and  $c(i,j)$  denotes the average resource usage by a unit of type  $(i,j)$  inbound traffic 14. An example of  $c(i,j)$  is the CPU cycles required to process one  $(i,j)$  type request. Finally,  $R_{\text{bound}}(i,j)$  denotes the absolute bound on  $R_t(i,j)$  when the ITS 20 computes new  $R_t(i,j)$ .

In accordance with the above, the following operations will be

completed during a service cycle time:

(a) The ITS unit 20 collects  $Ra(i,j)$ ,  $Rr(i,j)$ ,  $B(i,j)$  and optionally  $c(i,j)$ , and computes  $b(i,j)$ , the expected amount of output that would be generated when one unit of traffic type  $(i,j)$  is processed by a server 16. The ITS unit 20 also collects  
5  $Rbound(i,j)$  when available.

(b) The ITS unit 20 runs a rate scheduling (or freight load scheduling) algorithm to determine the best target values for  $Rt(i,j)$ . The ITS unit 20 may then compute  $Rt(i,j,k)$  if needed for each  $k^{th}$  server 16. The ITS unit 20 then relays  $Rt(i,j)$  values to one or more inbound traffic limiters (ITL) 22.

10 (c) The ITL 22 admits inbound traffic 14 at the rate  $Rt(i,j)$  in each service cycle time.

The inbound traffic dispatching network 12 has an inbound traffic monitor (ITM) 24 that observes the admitted traffic rates  $Ra(i,j)$  and the rejected traffic rates  $Rr(i,j)$ , and relays these rates to the ITS unit 20. Within the inbound  
15 traffic dispatching network 12, there could be more than one inbound traffic limiter (ITL) 22 and more than one inbound traffic monitor (ITM) 24. Although the inbound traffic monitor (ITM) 24 and inbound traffic limiter (ITL) 22 functions are shown and described as being associated with the inbound traffic dispatching network 12, these functions could be completely distributed to each individual server 16, as will be  
20 discussed below. Since the ITL 22 regulates the inbound traffic, it is convenient to put the inbound traffic monitoring functions at the ITL 22.

As also shown in Figure 2, each server 16 may have a resource usage monitor (RUM) 26 that observes server resource usage,  $c(i,j)$ , and an outbound traffic monitor (OTM) 28 that observes the outbound traffic,  $B(i,j)$ , both of which are relayed

to the ITS unit 20. There are a number of ways to observe the outbound traffic 18,  $B(i,j)$ , and any of which would be suitable for purposes of the present invention. The ITS unit 20 collects  $Ra(i,j)$ ,  $Rr(i,j)$ ,  $B(i,j)$  and optionally  $Rbound(i,j)$  and  $c(i,j)$ , and then computes the optimum values for  $Rt(i,j)$  that meet the service level agreements (SLAs) and relays these values to one or more ITLs 22. As represented in Figure 2, a server resource manager 21 is an optional means and its responsibility is to provide the absolute bound  $Rbound(i,j)$  on the rate  $Rt(i,j)$  regardless of the  $Bmax(i,j)$  given in the (min,max) SLAs.

Figures 3 and 4 schematically represent how the inbound traffic dispatching network 12 can be rendered highly scalable (large capacity) using existing dispatchers and a high-speed LAN (HS-LAN). In Figure 3, the inbound traffic limiting function and the inbound traffic monitoring function of the ITL 22 and ITM 24, respectively, are assigned to a standalone ITL unit 30, while in Figure 4 the inbound traffic limiting function and the inbound traffic monitoring function are assigned to each of a number of dispatchers 42, 44 and 46. With reference to Figure 3, the ITL unit 30 is connected to dispatchers 32, 34 and 36 via a high-speed LAN (HS-LAN) 31. The primary responsibility of the ITL unit 30 is to limit (thus dropping when needed) the inbound traffic (i,j) 14 by applying the target rates  $Rt(i,j)$  given by the ITS unit 20. While doing so, ITL unit 30 also monitors both admitted traffic  $Ra(i,j)$  and rejected traffic  $Rr(i,j)$ . Each dispatcher 32, 34 and 36 is responsible for dispatching (or load balancing) received traffic to associated servers 16 using any of its own load balancing algorithms. The traffic admittance algorithm used by the ITL 22 associated with the unit 30 for rate-based admittance is referred to as the rate-based inbound traffic regulation algorithm. While only one ITL unit 30 is represented in Figure 3, additional ITLs can be added to the high-speed LAN (HS-LAN) 31 to achieve even higher capacity, thus achieving higher scalability.

The inbound traffic dispatching network 12 of Figure 4 is structured

similarly to that of Figure 3, with a difference being that the inbound traffic limiting function and the inbound traffic monitoring function are assigned to each dispatcher 42, 44 and 46. Inbound traffic 14 are sent to dispatchers 42, 44 and 46 via a high-speed LAN (HS-LAN) 31. The dispatchers 42, 44 and 46 with the ITL functionality are responsible for regulating the inbound traffic 14 prior to dispatching traffic to the servers 16. In this embodiment, both ITL and ITM functionalities become the added functionalities to any existing dispatcher (or load balancing) units.

Figure 5 schematically represents the rate-based inbound traffic regulation algorithm executed by each ITL 22. This algorithm is repeated with each service cycle. Step 53 checks if the cycle-time has expired or not. If not expired, the algorithm moves to step 55. When the cycle-time has expired, the algorithm executes step 54, gets a new set of  $R_t(i,j)$  values if available, resets any other control and counter variables, and resets both  $R_a(i,j)$  and  $R_r(i,j)$  to zero for all  $i$  and  $j$ . Step 55 determines to which customer and traffic type  $(i,j)$  the received TCP connection request packet in step 50 belongs to so that a proper rate  $R_t(i,j)$  can be applied. In step 56, the algorithm checks whether or not the received TCP connection request packet of type  $(i,j)$  can be admitted by comparing  $R_a(i,j)$  against  $R_t(i,j)$ . In step 56, if  $R_a(i,j)$  is less than  $R_t(i,j)$ , the received TCP connection request packet is admitted by executing step 57. Step 57 increments  $R_a(i,j)$  by one and admits the packet. In step 56, if  $R_a(i,j)$  has reached  $R_t(i,j)$ , step 58 is executed. Step 58 increments  $R_r(i,j)$  by one and rejects (or drops) the received TCP connection request packet. Both step 57 and 58 lead to step 50. Step 50 gets a packet from inbound traffic 14. Step 51 checks whether or not the received packet is a TCP connection request. If not, the packet is simply admitted. If yes, step 53 is executed.

Figure 6 schematically represents an algorithm referred to above as the rate scheduling algorithm, which is executed by the ITS unit 20 to determine the optimum values for  $R_t(i,j)$  for all  $i$  and  $j$ . This scheduling algorithm starts at step 1

(61), which examines whether or not the service level agreements (SLAs) are all satisfied. Step 1 computes  $b(i,j)$  using the formula:

$$b(i,j) = a (B(i,j) / Ra(i,j)) + (1 - a) b(i,j)$$

where  $b(i,j)$  is the expected bandwidth usage per unit of inbound traffic 14,  $Ra(i,j)$  is the admitted inbound traffic,  $B(i,j)$  is the observed  $i^{\text{th}}$  customer's  $j^{\text{th}}$  type traffic total, and  $a$  is a value between 0 and 1.

Step 1 adjusts  $B_{\max}(i,j)$  by choosing the minimum of  $B_{\max}(i,j)$  itself and an optionally given bound  $R_{\text{bound}}(i,j) * b(i,j)$ . Here  $R_{\text{bound}}(i,j)$  is an "absolute bound" on  $R_t(i,j)$ . Since  $B_{\min}(i,j)$  must be less than or equal to  $B_{\max}(i,j)$ , this adjustment may affect to the value of  $B_{\min}(i,j)$  as well. Step 1 then computes  $B_t(i,j)$  and  $B_t$  and checks whether or not the generated outbound traffic is currently exceeding the total usable bandwidth  $B_{\text{total}}$  (that is detecting the outbound link congestion). If the congestion on the outbound link has been detected, step 2 (62) is executed. If there was no congestion detected and no packet dropping ( $R_r(i,j) = 0$ ) and no SLA has been violated, the algorithm moves to step 5 (65) and stops. Otherwise, step 1 moves to step 2 (62).

Step 2 (62) first computes the bandwidth requirement  $B_t(i,j)$  had no packets been dropped, that is the total inbound traffic ( $Ra(i,j) + R_r(i,j)$ ) for all  $(i,j)$  had been admitted. This bandwidth requirement  $B_t(i,j)$  could not exceed  $B_{\max}(i,j)$  and thus it is bounded by  $B_{\max}(i,j)$ . Step 2 then checks if the bandwidth requirements  $B_t(i,j)$  for all  $(i,j)$  can be supported without congesting the outbound link. If so, step 2 moves to step 4 (64) to convert the targeted bandwidth requirement to the targeted rates. If step 2 detects a possible congestion ( $B_t > B_{\text{total}}$ ), it then moves to step 3 (63) to adjust those  $B_t(i,j)$  computed in step 2 (62) so that the link level congestion could be avoided while guaranteeing the minimum bandwidth  $B_{\min}(i,j)$  for every  $(i,j)$ .

In step 3, two options are described: a first allows "bandwidth-borrowing" among customers, while in the second "bandwidth borrowing" among customers are not allowed. Here, "bandwidth borrowing" means letting some customers use the portion of the minimum guaranteed bandwidth not used by other customers. Step 3 first computes the "shareable" bandwidth. Step 3 then allocates (or prorates) the shareable bandwidth among those customer traffic classes that are demanding more than the guaranteed bandwidth  $B_{min}(i,j)$ . Although step 3 describes the use of "fair prorating of shareable bandwidth", this allocation discipline can be replaced by any other allocation discipline such as "weighted priority" or "weighted cost".

In step 4 (64), the bandwidth use targets  $Bt(i,j)$  computed in step 3 are converted to the target inbound traffic rates  $Rt(i,j)$ . When  $Bt(i,j)$  is less than or equal to the guaranteed minimum  $B_{min}(i,j)$ , there should be no "throttling" of the inbound traffic. Therefore,  $Bt(i,j)$  is set to  $B_{max}(i,j)$  for such  $(i,j)$  prior to converting  $Bt(i,j)$  to  $Rt(i,j)$ . In step 4, if the target rates are used by servers (as will be described later in Figure 7),  $Rt(i,j,k)$  must be computed from  $Rt(i,j)$  to balance the response time given by various servers 16 for each pair of  $(i,j)$  among all  $k$ . Doing so is equivalent to making the residual capacity or resource of all servers 16 equal, expressed by:

$$C(i,j,1) - Rt(i,j,1) c(i,j) = C(i,j,2) - Rt(i,j,2) c(i,j) = \dots C(i,j,n) - Rt(i,j,n) c(i,j) = d$$

where  $C(i,j,k)$  is the total resource allocated at server  $k$  for handling the traffic class  $(i,j)$ ,  $c(i,j)$  is the expected resource usage by a unit of  $(i,j)$  traffic and  $d$  is a derived value. Since

$$Rt(i,j) = \text{SUM of } Rt(i,j,k) \text{ for all } k = \text{SUM of } (C(i,j,k) - d) / c(i,j) \text{ for all } k$$

one can derive  $d$  from the above formula. Assuming a total of  $n$  servers:

- 15 -

$$d = (C(i,j) - R_t(i,j) c(i,j)) / n$$

where  $C(i,j)$  is the sum of  $C(i,j,k)$  over all  $k$ , and the formula for deriving  $R_t(i,j,k)$  from  $R_t(i,j)$  is

$$R_t(i,j,k) = (C(i,j,k) - (C(i,j) - R_t(i,j) c(i,j)) / n) / c(i,j)$$

5 Step 4 (64) leads to step 5 (65) and the rate scheduling algorithm stops.

Finally, Figure 7 represents a system in which the inbound traffic monitoring function (ITM) 70, inbound traffic limiting function (ITL) 72 and outbound traffic monitoring function (OTM) 74 are distributed to each server 16. Also distributed to each server 16 is resource use monitoring function (RUM) 76.

10 This system makes the inbound traffic dispatching network 12 in Figure 7 extremely simple. The inbound traffic dispatching network 12 of Figure 7 is very much like the one illustrated in Figure 4 except the dispatchers 42, 44 and 46 are simply replaced by dispatchers 32, 34 and 36. In this case, the ITS 20 executes the rate scheduling algorithm and derives  $R_t(i,j,k)$  from  $R_t(i,j)$  for every  $k$ . As in the case of the ITS 20 in  
 15 Figure 2, the ITS 20 in Figure 7 gets  $R_{\text{bound}}(i,j)$  from any server resource manager 21. The ITS 20 uses  $c(i,j)$ , the average of  $c(i,j,k)$  over  $k$ , in the derivation of  $R_t(i,j,k)$  from  $R_t(i,j)$ .  $c(i,j,k)$  are observed by the resource utilization monitoring function (RUM) 76 that resides in each server 16. Furthermore, each ITL 72 executes the rate-based inbound traffic regulation algorithm for  $(i,j,k)$  in place of  $(i,j)$  described in  
 20 reference to Figure 5. The ITS 20 relays  $R_t(i,j,k)$  values to each server  $k$ .

While the invention has been described in terms of a preferred embodiment, it is apparent that other forms could be adopted by one skilled in the art. Accordingly, the scope of the invention is to be limited only by the following claims.